PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of

	Prasenjit Sarkar	Art Unit:	2452
Serial No:	10/698,069	Examiner:	J. Chang
Filed:	October 30, 2003	Confirmation No.:	9724
For:	METHOD AND SYSTEM FOR INTERNET TRANSPORT ACCELERATION WITHOUT PROTOCOL OFFLOAD	Attorney Ref.:	ARC920030075US1

APPEAL BRIEF

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner For Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sirs:

This Appeal Brief is submitted in connection with the Notice of Appeal submitted February 2, 2009, the final Office Action dated October 20, 2008, and the Advisory Action dated November 20, 2008, in the above-captioned patent application.

REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation.

RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences that are related to the present appeal.

STATUS OF CLAIMS

Claim 7 is pending and stand finally rejected in the above-captioned patent application.

Claim 7 is the subject of this appeal.

STATUS OF AMENDMENTS

All amendments made to the claims have been entered. There are no amendments of the claims that have not been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

There is one (1) independent claim that is the subject of this appeal, independent claim 7. In the following paragraph, the references to the subject matter of independent claim 7 refer to locations in the originally filed patent application and the originally filed figures.

Independent claim 7 is directed to a method for direct data placement of data for an application that uses a network protocol. (See lines 1-3 of paragraph [01], lines 1-2 of paragraph [06], and lines 1-3 of paragraph [07].) According to claim 7, an application packet header is detected using a packet classifier (packet classifier 104, line 4 of paragraph [15] and Figure 1) within a network adapter (network adapter 103, line 3 of paragraph [15], and Figure 1). The application packet header belongs to a packet in a data stream associated with the application. Offsets included within the application header are identified (see lines 2-5 of paragraph [16]). A plurality of registers is loaded with the identified offsets (see lines 2-5 of paragraph [16], lines 2-4 of paragraph [18], and step 201 of Figure 2). Direct data placement of data associated with the application packet header is initiated when a result of masking a set of values corresponding to a direct data placement pattern with contents of the loaded registers matches at least one direct data placement pattern (see lines 10-14 of paragraph [18], and steps 205 and 206 of Figure 2). A plurality of direct data placement patterns is available for masking with the contents of the loaded registers (see lines 6-8 of paragraph [16], lines 6-10 of paragraph [18] and steps 204 and 205 of Figure 2.) Each direct data placement pattern is associated with an application packet header (see lines 6-8 of paragraph [16], lines 6-10 of

paragraph [18] and steps 204 and 205 of Figure 2). Initiating direct data placement comprises masking each of the plurality direct data placement patterns with contents of the loaded register (see lines 6-8 of paragraph [16], lines 6-10 of paragraph [18] and steps 204 and 205 of Figure 2.) Information corresponding to the detected application header is extracted (see lines 11-16 of paragraph [18]). A payload of the detected application header is mapped to a memory based on the direct data placement pattern, such that the memory is a predetermined region of memory reserved for the application (see lines 11-16 of paragraph [18]).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claim 7 stands finally rejected under 35 U.S.C. § 103(a) as unpatentable over Jolitz, U.S. Patent Application Publication No. 2001/0025315 A1, in view of Connery et al. (Connery), U.S. Patent No. 6,570,884 B1, and Follett et al. (Follett), U.S. Patent No. 6,094,712.

ARGUMENT

I. The Rejection Based On Jolitz in view of Connery and Follett

Applicants respectfully traverse this rejection. Applicants respectfully submit that the subject matter according to claim 7 is patentable over Jolitz in view of Connery and Follett.

Applicants respectfully submit that even if Jolitz, Connery and Follett are properly combinable, the resulting method is not the subject matter of claim 7.

In particular, none of Jolitz, Connery or Follett discloses or suggests the claimed memory that is a predetermined region of memory reserved for an application.

Further, none of Jolitz, Connery or Follett discloses or suggests a method comprising mapping a payload of the detected applications header to a memory based on the claimed direct data placement pattern, such that the claimed memory is a predetermined region of memory reserved for the application.

Lastly, none of Jolitz, Connery or Follett discloses or suggests a method comprising initiating direct data placement of data associated with the application packet header when a

result of masking a set of values corresponding to a direct data placement pattern with contents of the loaded registers matches at least one claimed direct data placement pattern.

A. None Of Jolitz, Follett Or Connery Discloses Or Suggests The Claimed Memory That Is A Predetermined Region Of Memory Reserved For An Application.

1. The Examiner admits that "Jolitz fails to teach that the memory is a predetermined region of memory reserved for the application." See final Office Action, page 3, lines 20-21.

2. Regarding Connery, the Examiner does not allege that Connery discloses or suggests the claimed memory that is a predetermined region of memory reserved for the application.

3. Applicants respectfully submit that Connery, in fact, does not disclose or suggest the claimed memory that is a predetermined region of memory reserved for the application according to claim 7.

4. At page 2, line 20, through page 3, line 7, of the Advisory Action dated November 20, 2008, the Examiner asserts

> "Applicant argues that Jolitz and Connery fail to teach that the memory is a predetermined region of memory reserved for an application. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The Office made not contention [sic] that Jolitz or Connery taught this limitation. As disclosed in the previous Office action, this limitation is taught by Follett."

5. Applicants respectfully submit that the Examiner misses the point being made by Applicants by demonstrating that neither Jolitz nor Connery disclose or suggest the claimed memory that is a predetermined region of memory reserved for the application. The

Advisory Action now makes explicitly clear that the Examiner is relying on Follett and only Follett as providing the claimed memory, whereas the conclusion of Examiner's syllogism (see, final Office Action, page 4, line 20, through page 4, line 7) does not make clear as to how the Examiner arrives at the claimed memory to form the conclusion. To address this uncertainty, Applicants have eliminated both Jolitz and Connery as providing the claimed memory of claim 7.[1] Consequently, by process of elimination, Applicants have logically demonstrated that this feature of claim 7 must be provided either (1) explicitly or implicitly by Follett, or (2) by some reasoning used by the Examiner that is based on Follett.[2]

6. Regarding Follett, at page 4, lines 3-7, of the final Office Action, the Examiner concludes that

> "It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to map a payload directly to a predetermined region of memory reserved for an application as taught by Follett in order to efficiently achieve protocol acceleration without intervention of the host operating system.

7. Applicants respectfully submit that the Examiner syllogism on which the Examiner forms this conclusion of obviousness is faulty.

8. In particular, the Examiner's major premise of the syllogism is "Jolitz fails to teach that the memory is a predetermined region of memory reserved for the application." (See final Office Action, page 3, lines 20-21.) Additionally, the Examiner has explicitly admitted that Connery does not disclose or suggest the claimed memory that is a predetermined region of memory reserved for the application according to claim 7. (See Advisory Action, page 3, lines 6-7.)

9. The minor premise of the Examiner's syllogism set forth at page 3, line 21, through page 4, line 3, of the final Office Action, is

---

[1] Applicants will also continue to take the position that neither Jolitz nor Connery disclose or suggest the claimed memory of claim 7.

[2] Applicants have demonstrated (above and elsewhere) that Follett buffer memory 120 is nothing but a buffer memory, which is not the claimed memory that is a predetermined region of memory reserved for the application. Thus, Follett does not directly provide the claimed memory, and it must be the Examiner's reasoning relating to Follett that is the basis for the combination of Jolitz, Follett and Connery to provide the claimed memory.

"Follett teaches mapping virtual addresses of applications to corresponding physical addresses, and locking the physical addresses to prevent inadvertent overwrite (Col. 4, lines 41-63). When data is received, the header is parsed, and the data is transferred to the specific physical memory locations that are mapped to the application virtual addresses (Col. 6, lines 18-33)."

10. Applicants respectfully submit that the Examiner's characterization of Follett ignores the fact that Follett discloses a dynamic buffer 120 that is used for buffering incoming data before the data is transferred to the memory associated with an application.

11. That is, Follett discloses an interface unit 54 that includes, among other components, a Virtual Circuit Identifier (VCI) memory map 95, a dynamic buffer memory 120 and a parser 139. VCI memory map 95 stores a map of the virtual addresses provided by an application 60 to the corresponding physical memory addresses in a main memory 44.[3] According to Follett, incoming data is loaded into buffer memory 120 when the incoming data is received. (See Follett, column 5, lines 18-21.) A parser 139 interprets the header of each ATM cell of the incoming data and causes a data portion of each ATM cell to be loaded into buffer memory 120. The incoming data is loaded into space within buffer memory 120 that is indexed by the VCI identified in the header of the ATM cell. (See Follett, column 5, lines 28-38.) Subsequently, the data is retrieved from buffer memory 120 and transferred to specific physical memory locations in main memory 44, as identified in VCI memory map 95. (See Follett, column 6, lines 18-24, and column 7, lines 6-12.)

[3] It is respectfully noted that at column 4, line 26, Follett makes reference to corresponding physical references in a memory 55; however, no memory 55 is depicted in any of the Figures of Follett. Instead, Figure 1B depicts a network interface unit 55 that is part of a disk server 35. It is also respectfully noted that Follett makes reference to application programs 48 that are contained in a main memory 44. (See Follett, column 2, lines 54-56.) But, Figure 1A depicts applications 60 that are contained in main memory 44 and column 4, lines 12-16, of Follett disclose an application 60 within main memory 44. Thus, Applicants respectfully submit that one of skill in the art (for whom the Follett specification was written) would conclude that the reference to application program 48 should be correctly interpreted as a reference to applications 60, and that an application 60 is contained in main memory 44. (Also see Follett, column 3, lines 55-65.) Further, Applicants respectfully submit that the reference to a memory 55 at column 4, line 26, of Follett would be interpreted by one of skill in the art to actually be a reference to memory 44.

12. Follett discloses each writing operation into buffer memory 120 is generally followed by a reading operation in which information is retrieved from buffer memory 120. Higher priority reads are recognized first, followed by lower priority retrievals. Variable blocks of data are transferred from buffer memory 120 to the host computer to optimize latency. When buffer memory 120 begins to saturate, larger blocks of data are transferred to the host computer, thereby optimizing transfer efficiency and preventing memory overload. Follett discloses that transferring larger blocks of data to the host computer ordinarily occurs as a result of host saturation so that access to the system bus is temporarily limited. (See Follett, column 7, lines 23-40, emphasis added.)

13. Thus, Applicants respectfully submit that Follett buffer memory 120 is, in actuality, no different from Jolitz application transfer memory 24.[4] That is, both the Follett buffer memory 120 and Jolitz application transfer memory 24 are used as temporary buffers to store received data before being transferred to a predetermined region of memory reserved for the application.

14. Nevertheless, the Examiner concludes that Follett provides the claimed memory that is a predetermined region of memory reserved for the application according to claim 7.

15. Consequently, Applicants respectfully submit that the logic used by the Examiner to conclude the Follett provides the claimed memory is unconvincing.

16. That is,

> "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). (See, also, MPEP §§ 706.02(j) and 2144.)

[4] Lines 1-2 of paragraph [0046] of Jolitz disclose that "[d]ata for the packets are buffered for transfer using the memory 24." [Emphasis added.]

17. In this case, the Examiner has not asserted that any of Jolitz, Follett or Connery explicitly provides the claimed memory. Further, the Examiner cannot rely on Follett as implicitly providing the claimed memory because Applicants have demonstrated that the Follett buffer memory 120 is not the claimed memory. Consequently, the Examiner must be relying on a convincing line of reasoning as to why an artisan would have found the claimed memory to have been obvious in light of the teachings of Follett.

18. As demonstrated, though, the logic used by the Examiner to rely on Follett as providing the claimed memory is faulty. Consequently, the Examiner has not presented a convincing line of reasoning as to why an artisan would have found the claimed memory of claim 7 to have been obvious in light of the teachings of the Follett.

19. Thus, none of Jolitz, Follett or Connery discloses or suggests the claimed memory that is a predetermined region of memory reserved for the application according to claim 7.

B. None Of Jolitz, Follett Or Connery Discloses Or Suggests Mapping A Payload Of The Detected Application Header To A Memory Based On The Direct Data Placement Pattern, Such That The Claimed Memory Is A Predetermined Region Of Memory Reserved For The Application.

1. As demonstrated above, the logic on which the Examiner relies for providing the claimed memory is faulty.

2. Moreover, Applicants respectfully submit that for the Examiner to also conclude that the applied art provides mapping a payload of the detected application header to a memory based on the direct data placement pattern, such that the claimed memory is a predetermined region of memory reserved for the application, is without basis. How can a payload of a detected application be mapped into a memory that is not provided by any of the applied art?

3. Further still, as demonstrated above, retrieval of data from buffer memory 120 to the host computer is based on a priority level of the incoming data and a latency

optimization to prevent memory overload, and is not mapping a payload of the detected application header to a memory based on the direct data placement pattern, such that the memory is a predetermined region of memory reserved for the application according to claim 7.

4. Thus, it follows that none of Jolitz, Follett or Connery discloses or suggests mapping a payload of the detected application header to a memory based on the claimed direct data placement pattern, such that the claimed memory is a predetermined region of memory reserved for the application.

C. None Of Jolitz, Follett Or Connery Discloses Or Suggests Initiating Direct Data Placement Of Data.

1. The Examiner asserts that paragraphs [0076] and [0075] of Jolitz discloses initiating direct data placement. (See final Office Action, page 3, lines 3-10.)

2. Applicants respectfully submit that, in actuality, paragraphs [0075] and [0076] of Jolitz discloses determining to which session an incoming IP datagram belongs. As previously pointed out, Jolitz discloses that once the session to which an incoming IP datagram belongs is determined, Rx engine 48 directly delivers a TCP payload to dual-port application transfer buffer (or memory) 24. (See Jolitz, paragraph [0044], lines 7-8.)

3. Further, in paragraphs [0058]-[0060], Jolitz discloses a data-receive sequence. As disclosed by Jolitz, receive data is placed in the proper bank of dual port application transfer memory 24 – the proper bank being either the shadow bank of application transfer memory 24 for the application bank of application transfer memory 24. If a packet is successfully received, the net payload data stored in the shadow bank of dual port application transfer memory 24 must be presented to the host system on a byte-by-byte level. If the packet is not successfully received, the state machine controlling the Jolitz device goes to an idle state and no changes occur to dual port application transfer memory 24.

4. Applicants respectfully submit that presenting a successfully received packet to the host system on a "byte-by-byte" level is certainly not direct data placement in accordance with the claimed subject matter of claim 7.

5. Thus, contrary to the Examiner's assertion, paragraphs [0075] and [0076] of Jolitz does not disclose or suggest initiating direct data placement.

6. Regarding Follett, the Examiner does not allege that Follett discloses or suggests initiating direct data placement of data.

7. Moreover, Applicants respectfully submit that Follett does not disclose or suggest a method comprising initiating direct data placement of data associated with the application packet header when a result of masking a set of values corresponding to a direct data placement pattern with contents of the loaded registers matches the claimed at least one direct data placement pattern.

8. Regarding Connery, the Examiner does not allege that Connery discloses or suggests initiating direct data placement of data.

9. Moreover, Applicants respectfully submit that Connery does not disclose or suggest a method comprising initiating direct data placement of data associated with the application packet header when a result of masking a set of values corresponding to a direct data placement pattern with contents of the loaded registers matches the claimed at least one direct data placement pattern.

10. Thus, none of Jolitz, Follett or Connery discloses or suggests a method comprising initiating direct data placement of data associated with the application packet header when a result of masking a set of values corresponding to a direct data placement pattern with contents of the loaded registers matches the claimed at least one direct data placement pattern.

CONCLUSION

In view of the above arguments, it is urged that the present application is in condition for allowance.

It is requested that this application be passed to issue with claim 7.

Respectfully submitted,

Date: February 2, 2009

Joseph P. Curtin
Registration No. 34,571
1469 N.W. Morgan Lane
Portland, Oregon 97229-5291
(503) 296-8373; (503) 297-0452 (facsimile)

CLAIMS APPENDIX

Claims 1-6. (canceled)

7. (previously presented) A method for direct data placement of data for an application that uses a network protocol, the method comprising:

detecting an application packet header using a packet classifier within a network adapter, the application packet header belonging to a packet in a data stream associated with the application;

identifying offsets included within the application header;

loading a plurality of registers with the identified offsets;

initiating direct data placement of data associated with the application packet header when a result of masking a set of values corresponding to a direct data placement pattern with contents of the loaded registers matches at least one direct data placement pattern, a plurality of direct data placement patterns being available for masking with the contents of the loaded registers, each direct data placement pattern being associated with an application packet header, and wherein initiating direct data placement comprises:

masking each of the plurality direct data placement patterns with contents of the loaded registers;

extracting information corresponding to the detected application header; and

mapping a payload of the detected application header to a memory based on the direct data placement pattern, the memory being a predetermined region of memory reserved for the application.

Claims 8-24. (canceled)

EVIDENCE APPENDIX

No Additional Evidence Submitted

RELATED PROCEEDINGS APPENDIX

No related proceedings